What is claimed is:

1. A method for counting performance events for a computing system, wherein the system includes i) a processor having a set of on-chip performance monitoring counter registers and ii) system memory, the method comprising the steps of:

a) incrementing an N-bit value in one of the performance monitoring counter registers responsive to occurrences of a monitored event for a first processing thread so that the N-bit value provides a measured value for the event in association with the thread;

b) merging the N-bit value of the counter register for the first processing thread into an N+M-bit value in an accumulator in system memory, wherein the merging is responsive to a switch from the first thread to a second thread; and

c) restoring the N-bit value for the first thread to the counter register from the accumulator responsive to a switch back to the first thread, wherein despite any thread switches the restoring maintains a value in the counter register for the first thread that is coherent relative to a previous value in the counter register for the first thread and the merging of the N-bit value into the accumulator maintains a coherent value for the larger, N+M-bit accumulated value.

2. The method of claim 1, wherein the counter register and the accumulator each have respective least-significant-bits segments and most-significant bits segments, and wherein step b) includes overwriting the least-significant-bits segment of the accumulator by the least-significant-bits segment of the counter, and adding the most-significant bits segment of the counter to the most-significant bits segment of the accumulator.

3. The method of claim 1, wherein step c) includes resetting the most-significant-bits segment of the counter and overwriting the least-significant-bits segment of the counter by the least-significant-bits segment of the accumulator.

4. The method of claim 1, comprising the step of performing an update-and-read operation responsive to a call to read the accumulator, the operation comprising the steps of:

reading a value of the counter in the user state;

reading a value of the accumulator in the system state;

merging the counter value with the accumulator value and saving this as an early-read-time performance event count;

reading a new value of the counter in the user state; and

merging the new counter value with the accumulator value and saving this value as a late-read-time performance event count.

5. The method of claim 1, comprising the step of computing the number of monitored events for a certain process, the computing comprising the steps of:

performing a first update-and-read operation responsive to a call to read the accumulator at a beginning of the certain process, wherein such an update-and-read operation comprises the steps of:

reading a value of the counter in the user state;

reading a value of the accumulator in the system state;

merging the counter value with the accumulator value and saving this as an early-read-time performance event count;

reading a new value of the counter in the user state; and

merging the new counter value with the accumulator value and saving this value as a late-read-time performance event count;

performing a second update-and-read operation responsive to a call to read the accumulator at an ending of the certain process;

calculating a difference between the late-read-time performance event count saved for the first update-and-read operation and the early-read-time performance event count saved for the second update-and-read operation.

6. The method of claim 5, wherein the counter register and the accumulator each have respective least-significant-bits segments and most-significant bits segments, and wherein step b) includes overwriting the least-significant-bits segment of the accumulator by the least-significant-bits segment of the counter, and adding the most-significant bits segment of the counter to the most-significant bits segment of the accumulator.

7. The method of claim 6, wherein step c) includes resetting the most-significant-bits segment of the counter and overwriting the least-significant-bits segment of the counter by the least-significant-bits segment of the accumulator.

8. An apparatus for counting performance events for a computing system, wherein the system includes i) a processor having a set of on-chip performance monitoring counter registers and ii) system memory, the apparatus comprising:

a) means for incrementing an N-bit value in one of the performance monitoring counter registers responsive to occurrences of a monitored event for a first processing thread so that the N-bit value provides a measured value for the event in association with the thread;

b) means for merging the N-bit value of the counter register for the first processing thread into an N+M-bit value in an accumulator in system memory, wherein the merging is responsive to a switch from the first thread to a second thread; and

c) means for restoring the N-bit value for the first thread to the counter register from the accumulator responsive to a switch back to the first thread, wherein despite any thread switches the restoring means maintains a value in the counter register for the first thread that is coherent relative to a previous value in the counter register for the first thread and the merging of the N-bit value into the accumulator maintains a coherent value for the larger, N+M-bit accumulated value.

9. The apparatus of claim 8, wherein the counter register and the accumulator each have respective least-significant-bits segments and most-significant bits segments, and wherein the merging means includes means for overwriting the least-significant-bits segment of the accumulator by the least-significant-bits segment of the counter and adding the most-significant bits segment of the counter to the most-significant bits segment of the accumulator.

10. The apparatus of claim 8, wherein the restoring means includes means for resetting the most-significant-bits segment of the counter and overwriting the least-significant-bits segment of the counter by the least-significant-bits segment of the accumulator.

11. The apparatus of claim 8, comprising means for performing an update-and-read operation responsive to a call to read the accumulator, the means for performing the operation comprising:

means for reading a value of the counter in the user state;

means for reading a value of the accumulator in the system state;

means for merging the counter value with the accumulator value and saving this as an early-read-time performance event count;

means for reading a new value of the counter in the user state; and

means for merging the new counter value with the accumulator value and saving this value as a late-read-time performance event count.

12. The apparatus of claim 8, comprising means for computing the number of monitored events for a certain process, the computing means comprising:

means for performing a first update-and-read operation responsive to a call to read the accumulator at a beginning of the certain process, wherein such a means for performing the operation comprises:

means for reading a value of the counter in the user state;

means for reading a value of the accumulator in the system state;

means for merging the counter value with the accumulator value and saving this as an early-read-time performance event count;

means for reading a new value of the counter in the user state; and

means for merging the new counter value with the accumulator value and saving this value as a late-read-time performance event count;

means for performing a second update-and-read operation responsive to a call to read the accumulator at an ending of the certain process; and

means for calculating a difference between the late-read-time performance event count saved for the first update-and-read operation and the early-read-time performance event count saved for the second update-and-read operation.

13. The apparatus of claim 12, wherein the counter register and the accumulator each have respective least-significant-bits segments and most-significant bits segments, and wherein the merging means includes means for overwriting the least-significant-bits segment of the accumulator by the least-significant-bits segment of the counter and adding the most-significant bits segment of the counter to the most-significant bits segment of the accumulator.

14. The apparatus of claim 13, wherein the restoring means includes means for resetting the most-significant-bits segment of the counter and overwriting the least-significant-bits segment of the counter by the least-significant-bits segment of the accumulator.

15. A computer program product for counting performance events for a computing system, wherein the system includes i) a processor having a set of on-chip performance monitoring counter registers and ii) system memory, the computer program product comprising:

a) instructions for incrementing an N-bit value in one of the performance monitoring counter registers responsive to occurrences of a monitored event for a first processing thread so that the N-bit value provides a measured value for the event in association with the thread;

b) instructions for merging the N-bit value of the counter register for the first processing thread into an N+M-bit value in an accumulator in system memory, wherein the merging is responsive to a switch from the first thread to a second thread; and

c) instructions for restoring the N-bit value for the first thread to the counter register from the accumulator responsive to a switch back to the first thread, wherein despite any thread switches the restoring means maintains a value in the counter register for the first thread that is coherent relative to a previous value in the counter register for the first thread and the merging of the N-bit value into the accumulator maintains a coherent value for the larger, N+M-bit accumulated value.

16. The computer program product of claim 15, wherein the counter register and the accumulator each have respective least-significant-bits segments and most-significant bits segments, and wherein the instructions for merging include instructions for overwriting the least-significant-bits segment of the accumulator by the least-significant-bits segment of the counter and adding the most-significant bits segment of the counter to the most-significant bits segment of the accumulator.

17. The computer program product of claim 15, wherein the instructions for restoring include instructions for resetting the most-significant-bits segment of the counter and overwriting the least-significant-bits segment of the counter by the least-significant-bits segment of the accumulator.

18. The computer program product of claim 15, comprising instructions for performing an update-and-read operation responsive to a call to read the accumulator, the instructions for performing the operation comprising:

instructions for reading a value of the counter in the user state;

instructions for reading a value of the accumulator in the system state;

instructions for merging the counter value with the accumulator value and saving this as an early-read-time performance event count;

instructions for reading a new value of the counter in the user state; and

instructions for merging the new counter value with the accumulator value and saving this value as a late-read-time performance event count.

19. The computer program product of claim 15, comprising instructions for computing the number of monitored events for a certain process, the instructions for computing comprising:

instructions for performing a first update-and-read operation responsive to a call to read the accumulator at a beginning of the certain process, wherein such instructions for performing the operation comprise:

instructions for reading a value of the counter in the user state;

instructions for reading a value of the accumulator in the system state;

instructions for merging the counter value with the accumulator value and saving this as an early-read-time performance event count;

instructions for reading a new value of the counter in the user state; and

instructions for merging the new counter value with the accumulator value and saving this value as a late-read-time performance event count;

instructions for performing a second update-and-read operation responsive to a second call to read the accumulator at an ending of the certain process; and

instructions for calculating a difference between the late-read-time performance event count saved for the first update-and-read operation and the early-read-time performance event count saved for the second update-and-read operation.

20. The computer program product of claim 19, wherein the counter register and the accumulator each have respective least-significant-bits segments and most-significant bits segments, and wherein the instructions for merging include instructions for overwriting the least-significant-bits segment of the accumulator by the least-significant-bits segment of the counter and adding the most-significant bits segment of the counter to the most-significant bits segment of the accumulator.

21. The computer program product of claim 20, wherein the instructions for restoring include instructions for resetting the most-significant-bits segment of the counter and overwriting the least-significant-bits segment of the counter by the least-significant-bits segment of the accumulator.